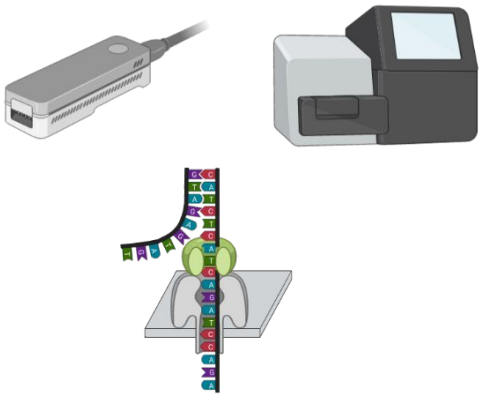
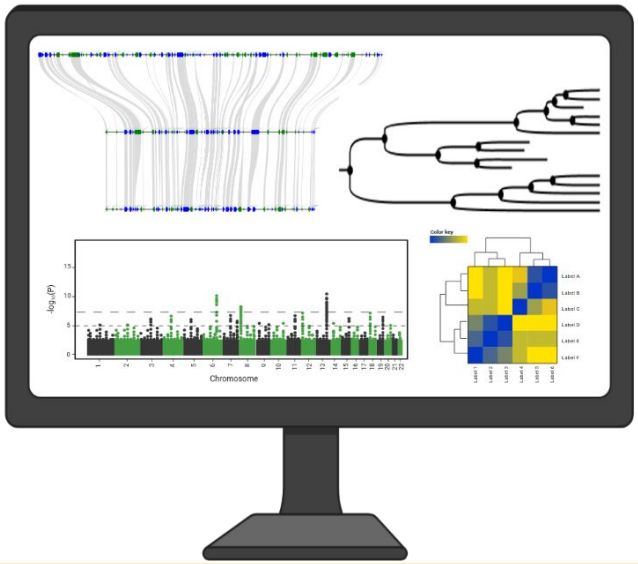
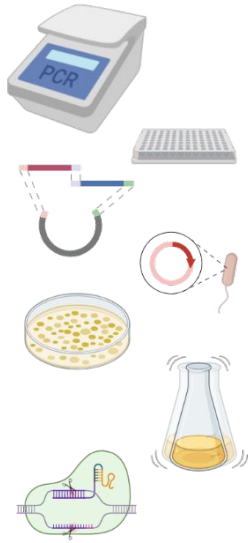
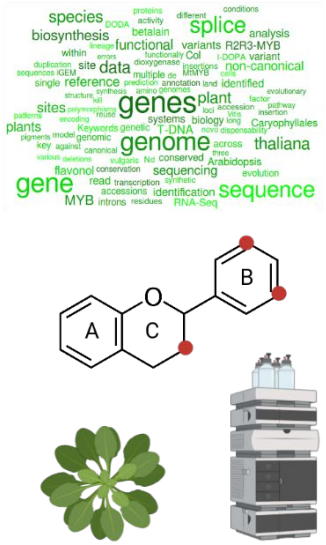
Technische Universität Braunschweig









Plant Biotechnology and Bioinformatics

Python - Control structures

Prof. Dr. Boas Pucker (Plant Biotechnology and Bioinformatics)

Availability of slides

- All materials are freely available (CC BY) - after the lectures:
  - StudIP: ‘Python for Life Scientists’
  - GitHub: https://github.com/bpucker/teaching

- Questions: Feel free to ask at any time

- Feedback, comments, or questions: b.pucker[a]tu-bs.de

If & else

- Distinguish between two cases:
  - Analyzing table: row of interest / irrelevant row

- Action depends on result of comparison

```
a = 5 #define variable
#user inputs number
b = int( input("please enter number!") )
if b < a:#if b is smaller than a
    print("b is smaller than a")
else:
    print("b is NOT smaller than a")
```

```
please enter number!3
b is smaller than a
```

elif

- Distinguish between multiple cases
  - Annealing temperature for PCR: suitable / too low / too high

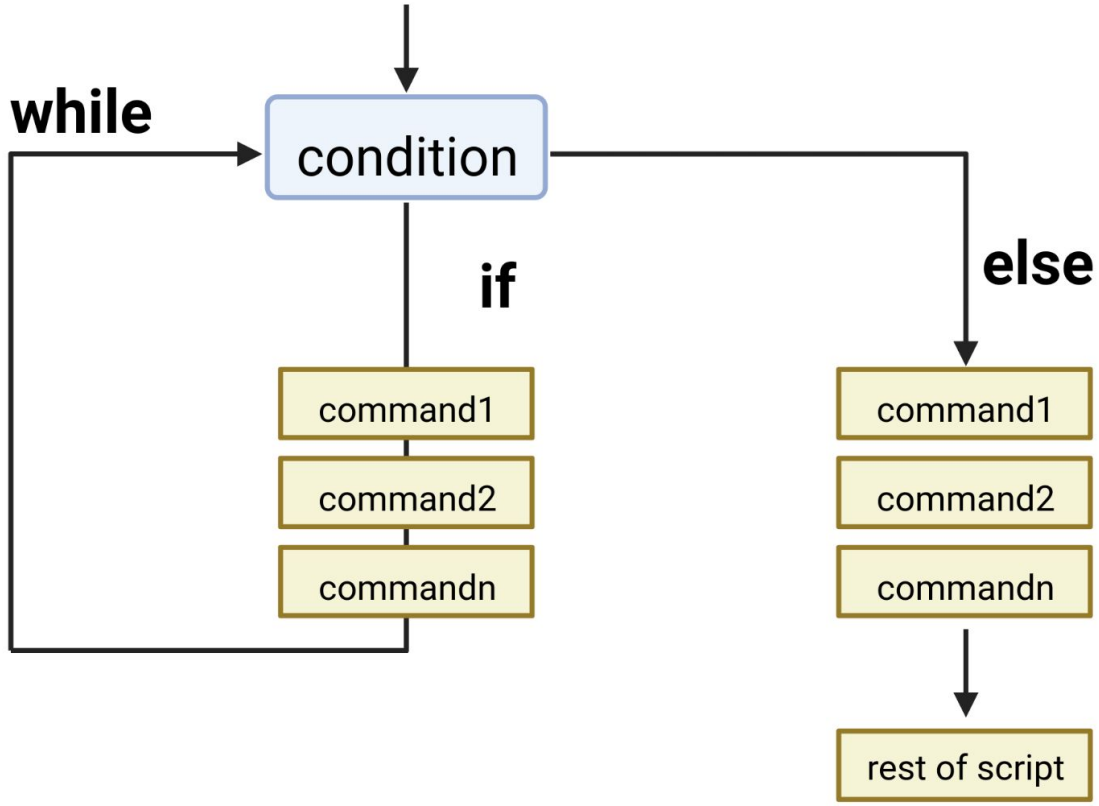
- Action depends on result of comparison

```
a = 5 #define variable
#user inputs number
b = int( input("please enter number!") )
if b < a:#if b is smaller than a
    print("b is smaller than a")
elif b == a:
    print("b is matching a")
else:
    print("b is bigger than a")
```

Exercises - Part3a

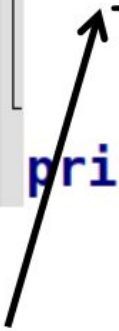
- 3.1) Write a script for guessing numbers!
- 3.2) Add tips (smaller/larger) during the guessing process!

Concept of loops



While loop (example)

```
1  a = 0
2  while a < 10:   #checks if a is smaller than 10
3      print(str( a ) + " is smaller than 10")
4      a += 1  #a = a+1
5      #something useful could happen here
6      print("a was increased by 1")
7  print(str( a ) + "is                10")
```



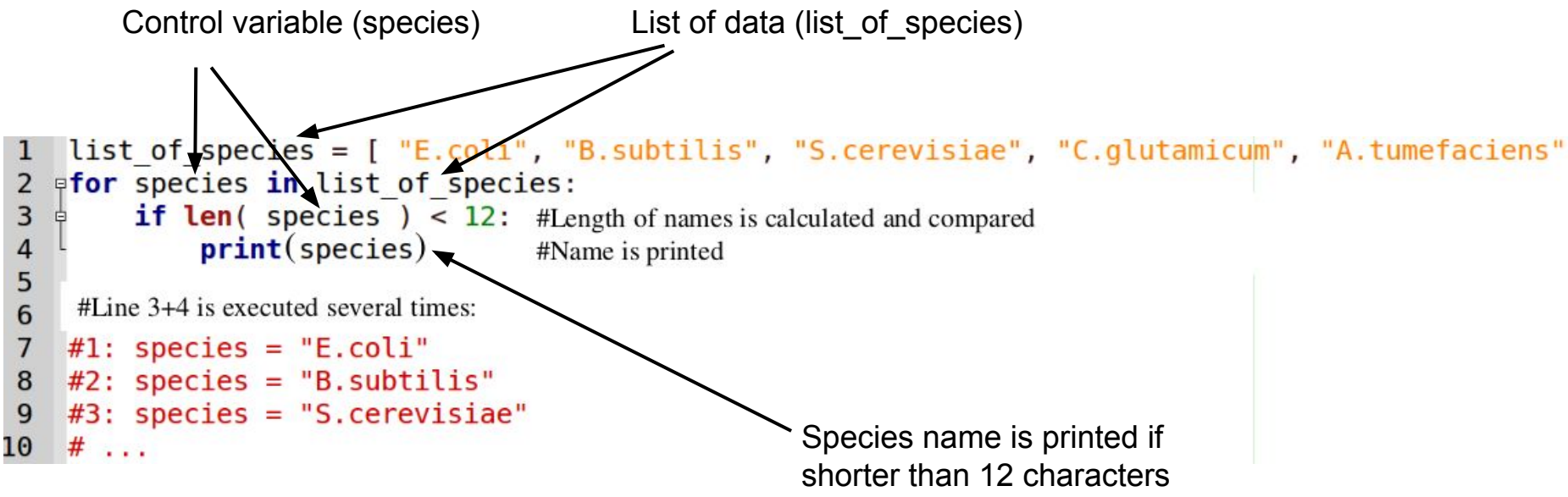
Code is executed until the condition for this loop becomes false

While infinite loop

WARNING: this loop is infinite!

```
1  #infinite loop:
2  a = 0
3  while True: #always true
4      a += 1  #a = a+1
5      print(str( a ))
6  print ("this line is never reached")
```

For loop


Control variable (species)
List of data (list_of_species)
1 list_of_species = ["E.coli", "B.subtilis", "S.cerevisiae", "C.glutamicum", "A.tumefaciens"
2 for species in list_of_species:
3 if len(species) < 12: #Length of names is calculated and compared
4 print(species) #Name is printed
5
6 #Line 3+4 is executed several times:
7 #1: species = "E.coli"
8 #2: species = "B.subtilis"
9 #3: species = "S.cerevisiae"
10 # ...
Species name is printed if shorter than 12 characters

Exercises - Part3b

- 3.3) Write a function counting to 100 and printing all number which can be divided by 4 without any residue!
  - Info: 10%2 #modulo division in Python

- 3.4) Write a function counting down from 1000 to 0 and printing all numbers!

- 3.5) Generate a list of species names! Write a function that prints all species names starting with ‘E’!

- 3.6) Expand this function to limit the printing to species names which are additionally shorter than 10 characters!

- 3.7) Expand this function to limit the printing to species names which are additionally ending with ‘a’.

Exercises - Part3b (addition)

- 3X1: Print species names that match at least 2 of the following 3 conditions: (1) longer than 8 characters, (2) start with ‘E’, (3) end with ‘a’

- 3X2: Print species names that are identical to the previous or next species name in the list.

- 3X3: Implement a function that can quickly sort the elements of a list alphabetically.

range()

```
1  list_of_species = ["E.coli", "B.subtilis", "S.cerevisiae", "C.glutamicum", "A.tumefaciens"]
2  length = len( list_of_species ) #length = 5
3  for i in range( length ):   #starts at 0 and runs to i=4 (five values)
4      if len( list_of_species[ i ] ) < 12:    #length of name is calculated and compared
5          print(list_of_species[ i ]) #name is printed
6
7  #i is taking five different values:
8  #1: i=0
9  #2: i=1
10 #3: i=2
11 #4: i=3
12 #5: i=4
13 #i=5 is never reached by range()
```

enumerate()

```
1  list_of_species = ["E.coli", "B.subtilis", "S.cerevisiae", "C.glutamicum", "A.tumefaciens"]
2  for idx, species in enumerate( list_of_species ):
3      if len( species ) < 12: #length of names is calculated and compared
4          print("position of " + species + " is: " + str( idx ))
5
6  #i is taking five different values:
7  #1: i=0 and species="E.coli"
8  #2: i=1 and species="B.subtilis"
9  #3: i=2 and species="S.cerevisiae"
10 #4: i=3 and species="C.glutamicum"
11 #5: i=4 and species="A.tumefaciens"
12 #i=5 is never reached
```

Exercises - Part3c

- 3.8) Write a script to print 50x ‘here’ and the current value of the control variable!
- 3.9) Write a script to walk through the species list and to print the character from the species where the index corresponds to the current control variable value!

Checking a list/string for a sublist/substring

- Check if an element is contained in a list:

```
genes = ["g1", "g2", "g3", "g4", "g5"]
if "g3" in genes:
    print("g3 in genes")
```

```
g3 in genes
```

- Check if a substring is contained in a string:

```
genome_seq = "ACACCGATTACTGGAGGTTACGTAATGGCCA"
primer = "ACTGGAGGTTACG"
if primer in genome_seq:
    print("primer binding site detected.")
else:
    print("no primer binding site detected")
```

```
primer binding site detected.
```

Break & pass

- Break allows you to exit a loop once a condition is met
- Pass allows you to skip elements in a loop

```
1 values = ["species1", "species2", "species3", "species4"]
2 for val in values:
3     if int( val[-1] ) < 3:
4         pass
5     elif int( val[-1] ) == 4:
6         break
```

Dictionaries

- Dictionary = data structure that allows very efficient access
- update() = adds new elements (dictionaries) to an existing dictionary

```
1 # dictionary:
2 my_dictionary = { 'key1': 'valueA', 'key2': 'valueB', 'key3': 'valueC' }
3
4 # adding an element to a dictionary:
5 my_dictionary.update( { 'key4': 'valueD' } )
```

Try & except

- Try = run a code block until an error occurs
- Except = capture one or multiple errors and run alternative code

```
#Try & except:
my_dictionary = { 'key1': 'valueA', 'key2': 'valueB', 'key3': 'valueC' }

dict_keys = list( my_dictionary.keys() )
for key in dict_keys:
    try:
        value = my_dictionary[ key ]
    except KeyError:
        print( "invalid key: " + key )
```

Time for questions!